Flex static code analysis

Unique rules to find Bugs, Security Hotspots, and Code Smells in your FLEX code

Delivering code in production with debug features activated is security-sensitive

Analyze your code

Security Hotspot | Minor | cwe error-handling debug user-experience owasp

Delivering code in production with debug features activated is security-sensitive. It has led in the past to the following vulnerabilities:

- CVE-2018-1999007
- CVE-2015-5306
- CVE-2013-2006

An application's debug features enable developers to find bugs more easily and thus facilitate also the work of attackers. It often gives access to detailed information on both the system running the application and users.

Ask Yourself Whether

- the code or configuration enabling the application debug features is deployed on production servers.
- the application runs by default with debug features activated.

There is a risk if you answered yes to any of those questions.

Recommended Secure Coding Practices

Do not enable debug features on production servers.

Sensitive Code Example

```
if (unexpectedCondition)
{
  Alert.show("Unexpected Condition"); // Sensitive
}
```

The `trace()` function outputs debug statements, which can be read by anyone with a debug version of the Flash player:

```
var val:Number = doCalculation();
trace("Calculation result: " + val);  // Sensitive
```

See

- OWASP Top 10 2017 Category A3 - Sensitive Data Exposure
- MITRE, CWE-489 - Leftover Debug Code
- MITRE, CWE-215 - Information Exposure Through Debug Information

Available In:

sonarcloud | sonarqube

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.
Privacy Policy

override "Event.clone()"
Bug

Constructors should not dispatch events

Bug

"ManagedEvents" tags should have companion "Event" tags

Bug

Objects should not be instantiated inside a loop

Code Smell

Two branches in a conditional structure should not have exactly the same implementation

Code Smell

Constructor bodies should be as lightweight as possible

Code Smell

Only "while", "do" and "for" statements should be labelled

Code Smell

Statements, operators and keywords specific to ActionScript 2 should not be used

Code Smell

"for" loop stop conditions should be invariant

Code Smell

Unused function parameters should be removed

Code Smell